



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/692,765	10/19/2000	Thomas E. Saulpaugh	5181-65700	8734
7590	04/08/2004		EXAMINER	
Robert C Kowert Conley Rose & Tayon PC P O Box 398 Austin, TX 78767			PATEL, ASHOKKUMAR B	
			ART UNIT	PAPER NUMBER
			2154	10
DATE MAILED: 04/08/2004				

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	Application No.	Applicant(s)
	09/692,765	SAULPAUGH ET AL. <i>JW</i>
	Examiner	Art Unit
	Ashok B. Patel	2154

~ The MAILING DATE of this communication appears on the cover sheet with the correspondence address ~  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) Responsive to communication(s) filed on \_\_\_\_.
- 2a) This action is FINAL.                                    2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) Claim(s) 1-48 is/are pending in the application.
  - 4a) Of the above claim(s) \_\_\_\_ is/are withdrawn from consideration.
- 5) Claim(s) \_\_\_\_ is/are allowed.
- 6) Claim(s) 1-48 is/are rejected.
- 7) Claim(s) \_\_\_\_ is/are objected to.
- 8) Claim(s) \_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on \_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.
 

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
  - a) All    b) Some \* c) None of:
    1. Certified copies of the priority documents have been received.
    2. Certified copies of the priority documents have been received in Application No. \_\_\_\_.
    3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. ____.
3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date <u>14</u> .	5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)
	6) <input type="checkbox"/> Other: ____.

## **DETAILED ACTION**

1. Application Number 09/692, 765 was filed on 10/19/2000. Claims 1-48 are subject to examination.

### ***Claim Rejections - 35 USC § 101***

2. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

- a. Claims 36-48 recite carrier medium and are failing to recite its computer readability. Amending the claims by addition of the phrase "computer readable carrier medium" will overcome the rejection.

### ***Claim Rejections - 35 USC § 102***

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1, 13, 14, 26, 36 and 48 are rejected under 35 U.S.C. 102(e) as being anticipated by Bowman-Amuah (US 6,289,382).

### **Referring to claims 1 and 13,**

The reference Bowman-Amuah teaches the concepts which are of a paramount importance as stated below such as publish and subscribe for a distributed computing environment and, how to access and manage data in Web documents so as to gain

more control over document structure, and as an answer to this question, it teaches, XML (a data representation language) which is a meta-language that allows authors to create their own customized tags to identify different types of data on their Web pages. In addition to improving document structure, these tags will make it possible to more effectively index and search for information in databases and on the Web. (col. 41, lines 1-14). The reference also teaches the interfaces having a unique set of services associated with it. The names of the interfaces are broadcasted to a plurality of systems requiring service. (col. 2, lines 21-40). The reference also teaches the Message-Oriented Middleware products typically support communication among various computing platforms (e.g., DOS, Windows, OS/2, Macintosh, UNIX, and mainframes) and one of them is publish and subscribe (also known as Push messaging)--as shown in FIG. 19, which is a special type of data delivery mechanism that allows processes to register an interest in (i.e., subscribe to) certain messages or events. An application then sends (publishes) a message, which is then forwarded to all processes that subscribe to it. (col. 69, lines 15-45). (wherein receiving in message in a data representation language sent to a client platform in the distributed computing environment from a service in the distributed computing environment, wherein the message includes a data representation language representation of an event generated by the service; and sending the data representation language representation of the event to one or more processes registered to receive the event from the service.)

**Referring to claims 14 and 26,**

Claims 14 and 26 are claims to the device configured for carrying out the steps of method of claims 1 and 13. Therefore, claims 14 and 26 are rejected for the reasons set forth for claims 1 and 13 and the reference Bowman-Amuah teaching of processor (Fig. 1, element 110), the memory coupled to processor (Fig. 1, elements 114 and 116) and the reference Bass's teaching of message broker (an event message gate unit) (Fig. 1, element 108) providing secure message endpoint.

**Referring to claims 36 and 48,**

Claims 36 and 48 are claims to a carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claims 1 and 13. Therefore, claims 36 and 48 are rejected for the reasons set forth for claims 1 and 13.

***Claim Rejections - 35 USC § 103***

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 2-12, 15-25, 27-35, and 37-47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bowman-Amuah (US 6,289,382) in view of Bass et al. (hereinafter Bass) (US 6,405,266)

**Referring to claims 2, 10 , 11 and 12,**

Keeping in mind the teachings of the reference Bowman-Amuah , the reference Bowman-Amuah also teaches the interfaces having a unique set of services associated with it (a message interface for a set of events generated by the service). The names of the interfaces are broadcasted to a plurality of systems requiring service. (col.2, lines 21-40). The reference fails to teach receiving a data representation language schema (a message interface) on the client platform. The reference Bass teaches of an interface (Fig. 1, element 108)(message broker) which is capable of receiving, delivering event messages in such a way that it is internally republished to subscribing processes (wherein the one or more processes are executing within the client platform.) The process can subscribe to an event type via the message broker.(col. 3, lines 64-66). (generating an event message end point for client platform) . The reference Bass also teaches that the invention can be implemented as software written in an object oriented language such as JAVA. (wherein the event is a Java event ). (col. 6, lines 38-39). Thereby the reference teaches generating an event message endpoint for the client platform wherein receiving a message and sending the data representation language representation of the event to one or more processes are performed by the event message endpoint. Therefore, it would have been obvious to one having ordinary skill in the art at the time of invention was made to combine Bowman-Amuah's generation of the interfaces having a unique set of services associated with each interfaces (a message interface for a set of events generated by the service) and delivering as part of the Bass's event messages in such a way that it is internally republished to subscribing processes(receiving the data representation language schema of the service in a

service advertisement of the service.). The advantage of this communication mechanism is that it does not burden each object with communication information and functionality. Moreover, the mechanism would provide location transparency for both local and remote destinations as taught by Bass.

**Referring to claims 3 and 7,**

The reference Bowman-Amuah teaches the interfaces having a unique set of services associated with it (a message interface for a set of events generated by the service). The names of the interfaces are broadcasted to a plurality of systems requiring service. (col.2, lines 21-40). The reference also teaches the Message-Oriented Middleware products typically support communication among various computing platforms (e.g., DOS, Windows, OS/2, Macintosh, UNIX, and mainframes) and one of them is publish and subscribe (also known as Push messaging)--as shown in FIG. 19, which is a special type of data delivery mechanism that allows processes to register an interest in (i.e., subscribe to) certain messages or events. An application then sends (publishes) a message, which is then forwarded to all processes that subscribe to it. (col. 69, lines 15-45). This teaching by the reference as a mechanism is of a paramount importance. The reference fails to teach the event message endpoint subscribing to one or more of the set of events generated by the service and each of the one or more processes registering interest in one or more of the set of events generated by the service with the event message endpoint subsequent to said generating an event message endpoint. The reference Bass teaches of an interface (Fig. 1, element 108)(message broker) which is capable of receiving, delivering event messages in such a way that it is

internally republished to subscribing processes. The process can subscribe to an event type via the message broker.(col. 3, lines 64-66). (an event message end point for client platform) .Thereby the reference teaches the event message endpoint subscribing to one or more of the set of events generated by the service, wherein the service is configured to send messages including data representation language representations of an event to subscribers to the event when the event is generated and each of the one or more processes registering interest in one or more of the set of events generated by the service with the event message endpoint subsequent to said generating an event message endpoint, which is in the publishing event message endpoint. Therefore, it would have been obvious to one having ordinary skill in the art at the time of invention was made to combine Bowman-Amuah's generation of the interfaces having a unique set of services associated with each interfaces and delivering as part of the Bass's event messages in such a way that it is internally republished to subscribing processes through the event message endpoint (message broker). The advantage of this communication mechanism is that it does not burden each object with communication information and functionality. Moreover, the mechanism would provide location transparency for both local and remote destinations as taught by Bass.

**Referring to claim 4,**

Keeping in mind the teachings of the reference, the reference Bowman-Amuah teaches the interfaces having a unique set of services associated with it (a message interface for a set of events generated by the service). The names of the interfaces are broadcasted to a plurality of systems requiring service. (col.2, lines 21-40). The reference also

teaches the Message-Oriented Middleware products typically support communication among various computing platforms (e.g., DOS, Windows, OS/2, Macintosh, UNIX, and mainframes) and one of them is publish and subscribe (also known as Push messaging)--as shown in FIG. 19, which is a special type of data delivery mechanism that allows processes to register an interest in (i.e., subscribe to) certain messages or events. An application then sends (publishes) a message, which is then forwarded to all processes that subscribe to it. (col. 69, lines 15-45). This teaching by the reference as a mechanism is of a paramount importance. The reference also teaches that an authentication can occur through various means such as, related to authentication, non-repudiation is a means of tagging a message. (col. 82, lines 19-22). The reference also teaches that XML is a meta-language that allows authors to create their own customized tags to identify different types of data and, XML documents consist of two parts. The first is the document itself, which contains XML tags for identifying data elements and resembles an HTML document. The second part is a DTD that defines the document structure by explaining what the tags mean and how they should be interpreted. (col. 41, lines 10-24). Thereby, the reference teaches that the data representation language message from the service includes an authentication credential for the service (XML message with authentication tag), the method further comprising the process using the authentication credential for the service to authenticate the data representation language message as being from the service (XML message with authentication tag and identifying the interface service). The reference fails to teach the event message endpoint. The reference Bass teaches of an interface (Fig. 1, element

108)(message broker) which is capable of receiving, delivering event messages in such a way that it is internally republished to subscribing processes. The process can subscribe to an event type via the message broker.(col. 3, lines 64-66). (an event message end point) . Therefore, it would have been obvious to one having ordinary skill in the art at the time of invention was made to combine Bowman-Amuah's teachings regarding authentication as stated above can be included as a tag in the XML message such that the Bass's message broker (event message endpoint) can authenticate to authenticate the data representation language message (XML message with authentication tags) as being from the service. Because, when a user requests access to network resources, the authorization service determines if the user has the appropriate permissions and either allows or disallows the access. This occurs after the user has been properly authenticated as taught by Bowman-Amuah.

**Referring to claims 5 and 6,**

In addition to the above, the reference Bowman-Amuah teaches the interfaces having a unique set of services associated with it (a message interface for a set of events generated by the service). The names of the interfaces are broadcasted to a plurality of systems requiring service. (col.2, lines 21-40). ( wherein the data representation language schema defines a set of messages that the service may send to the event message endpoint). The reference also teaches the Message-Oriented Middleware products typically support communication among various computing platforms (e.g., DOS, Windows, OS/2, Macintosh, UNIX, and mainframes) and one of them is publish and subscribe (also known as Push messaging)--as shown in FIG. 19, which is a

special type of data delivery mechanism that allows processes to register an interest in (i.e., subscribe to) certain messages or events. An application then sends (publishes) a message, which is then forwarded to all processes that subscribe to it. (col. 69, lines 15-45). This teaching by the reference as a mechanism is of a paramount importance. The reference Bowman-Amuah teaches environment verification services ensuring functionality by monitoring, identifying and validating environment integrity prior and during program execution such as, correct version. These services are invoked when an application begins processing or when a component is called. Applications can use these services to verify that the correct versions of required execution architecture components and other application components are available.(col. 99, lines 19-27).(verifying type correctness of the data representation language message according to the data representation language schema subsequent to said receiving a message). The reference fails to teach the event message endpoint. The reference Bass teaches of an interface (Fig. 1, element 108)(message broker) which is capable of receiving, delivering event messages in such a way that it is internally republished to subscribing processes. The process can subscribe to an event type via the message broker.(col. 3, lines 64-66). (an event message end point) . Therefore, it would have been obvious to one having ordinary skill in the art at the time of invention was made to combine Bowman-Amuah's teachings regarding verification of correct version under environment verification services as stated above can be done by the Bass's message broker (event message endpoint) Applications (processes) can use these services to verify that the

correct versions of required execution architecture components and other application components are available to them as taught by Bowman-Amuah.

**Referring to claims 8 and 9,**

Keeping in mind the teachings of Bowman-Amuah as stated above, the reference teaches the interfaces having a unique set of services associated with it. The names of the interfaces are broadcasted to a plurality of systems requiring service. (col.2, lines 21-40). The reference also teaches the Message-Oriented Middleware products typically support communication among various computing platforms (e.g., DOS, Windows, OS/2, Macintosh, UNIX, and mainframes) and one of them is publish and subscribe (also known as Push messaging)--as shown in FIG. 19, which is a special type of data delivery mechanism that allows processes to register an interest in (i.e., subscribe to) certain messages or events. An application then sends (publishes) a message, which is then forwarded to all processes that subscribe to it. (col. 69, lines 15-45). Thus the reference suggests by impicature that a process unregistering interest in a first event of the service; and the event message gate unsubscribing to the first event with the service subsequent to said unregistering; wherein the service is further configured to not send messages including data representation language representations of the first event to event message endpoints that are unsubscribed to the first event. However, the reference fails to teach the processes providing an event handler call method to the event message endpoint. The reference Bass teaches the one message broker for each process, which is internal to the process, and which will handle the distribution of events within the process. An object that is internal to the

process may subscribe to an event type via the message broker. When another object publishes an event with the same event type, the message broker will invoke a callback on the subscribing object in a thread safe manner. (col. 3, lines 62-67 and col. 4, lines 1-3). As such, the reference explains that a message broker API 108, 113 that is internal to each process to send and receive messages (or events) with the objects in the process. An object may subscribe to an event type via the message broker API. When another object within the process publishes an event with the same event type, the message broker will invoke a callback on the subscribing object in a thread safe manner. This will be repeated for all subscribing objects. Messages will be delivered only once and asynchronously. (col.5, lines 22-30). (wherein registering interest in one or more of the set of events comprises each of the one or more processes providing an event handler callback method to the event message endpoint; wherein said sending the data representation language representation of the event to one or more processes registered to receive the event from the service comprises: the event message endpoint calling an event handler method of each process registered with the event message endpoint to the event; and the event message endpoint passing the data representation language representation of the event to each called event handler.) Therefore, it would have been obvious to one having ordinary skill in the art at the time of invention was made to combine Bowman-Amuah's teachings regarding the interfaces having a unique set of services associated with it and the names of the interfaces are broadcasted to a plurality of systems requiring service as stated above with the Bass's message broker (event message endpoint) of each processes invoking the callback method to pass the

event. As such, since, events are delivered in their own thread of execution (such the publisher does not wait for the event to be delivered), then a single event delivery failure will not disable the entire system. Moreover, there is no need for objects that publish and subscribe to know about other objects within the process, i.e. there is no need to have a reference between the publisher and subscriber. Therefore, the message broker removes the burden from the objects of having the knowledge and functionality for determining internal/external objects, as well as the knowledge of object destination and PUB/SUB vendor as taught by Bass.

**Referring to claims 15, 18, 24 and 25,**

Claims 15, 18, 24 and 25 are claims to the device configured for carrying out the steps of method of claims 2, 10 , 11 and 12. Therefore, claims 15, 18, 24 and 25 are rejected for the reasons set forth for claims 2, 10 , 11 and 12 and the reference Bass's teaching of message broker (an event message gate unit) (Fig.1, element 108) providing secure message endpoint.

**Referring to claims 16 and 17,**

Claims 16 and 17 are claims to the device configured for carrying out the steps of method of claims 5 and 6. Therefore, claims 16 and 17 are rejected for the reasons set forth for claims 5 and 6 and the reference Bass's teaching of message broker (an event message gate unit) (Fig.1, element 108) providing secure message endpoint.

**Referring to claims 19 and 21,**

Claims 19 and 21 are claims to the device configured for carrying out the steps of method of claims 3 and 7. Therefore, claims 19 and 21 are rejected for the reasons set

forth for claims 3 and 7 and the reference Bass's teaching of message broker (an event message gate unit) (Fig.1, element 108) providing secure message endpoint.

**Referring to claim 20,**

Claim 20 is a claim to the device configured for carrying out the steps of method of claim 4. Therefore, claim 20 is rejected for the reasons set forth for claim 4 and the reference Bass's teaching of message broker (an event message gate unit) (Fig.1, element 108) providing secure message endpoint.

**Referring to claims 22 and 23,**

Claims 22 and 23 are claims to the device configured for carrying out the steps of method of claims 8 and 9. Therefore, claims 22 and 23 are rejected for the reasons set forth for claims 8 and 9 and the reference Bass's teaching of message broker (an event message gate unit) (Fig.1, element 108) providing secure message endpoint.

**Referring to claims 27, 28 and 35,**

The reference Bowman-Amuah teaching of processor (Fig. 1, element 110) , the memory coupled to processor (Fig. 1, elements 114 and 116). The reference Bowman-Amuah teaches the concepts which are of a paramount importance as stated below such as publish and subscribe for a distributed computing environment and, how to access and manage data in Web documents so as to gain more control over document structure, and as an answer to this question, it teaches, XML (a data representation language) which is a meta-language that allows authors to create their own customized tags to identify different types of data on their Web pages. In addition to improving document structure, these tags will make it possible to more effectively index and

search for information in databases and on the Web. (col. 41, lines 1-14). The reference also teaches the interfaces having a unique set of services associated with it. The names of the interfaces are broadcasted to a plurality of systems requiring service. (col. 2, lines 21-40). The reference also teaches the Message-Oriented Middleware products typically support communication among various computing platforms (e.g., DOS, Windows, OS/2, Macintosh, UNIX, and mainframes) and one of them is publish and subscribe (also known as Push messaging)--as shown in FIG. 19, which is a special type of data delivery mechanism that allows processes to register an interest in (i.e., subscribe to) certain messages or events. An application then sends (publishes) a message, which is then forwarded to all processes that subscribe to it. (col. 69, lines 15-45). (a processor; a memory coupled to said processor; a service process configured to generate an event, generate a message in a data representation language, wherein the message includes a data representation language representation of the event generated by the service process; and send the message to one or more event message gate units in the distributed computing environment;). The reference fails to teach event message gate units and wherein each of the one or more event message gate units are operable to distribute the data representation language representation of the event sent in the message from the service process to one or more processes registered to receive the event from the service process. The reference Bass teaches the message broker (Fig. 1, element 108) which has the functionality of event message gate unit and service message gate unit and, the distribution of the data representation language representation of the event by the message broker as described in Abstract.

Therefore, it would have been obvious to one having ordinary skill in the art at the time of invention was made to combine Bowman-Amuah's generation of the interfaces having a unique set of services associated with each interfaces and delivering as part of the Bass's event messages in such a way that it is internally republished to subscribing processes through the event message endpoint (message broker). The advantage of this communication mechanism is that it does not burden each object with communication information and functionality. Moreover, the mechanism would provide location transparency for both local and remote destinations as taught by Bass.

**Referring to claims 29, 31 and 34,**

Claims 29, 31 and 34 are claims to the device configured for carrying out the steps of method of claims 2, 10 and 12. Therefore, claims 29, 31 and 34 are rejected for the reasons set forth for claims 2, 10 , 11 and 12 and the reference Bass's teaching of message broker (an event message gate unit) (Fig.1, element 108) providing secure message endpoint.

**Referring to claim 30,**

Claim 30 is a claim to the device configured for carrying out the steps of method of claim 6. Therefore, claim 30 is rejected for the reasons set forth for claims 5 and 6 and the reference Bass's teaching of message broker (an event message gate unit) (Fig.1, element 108) providing secure message endpoint.

**Referring to claim 32,**

Claim 32 is a claim to the device configured for carrying out the steps of method of claim 3. Therefore, claim 32 is rejected for the reasons set forth for claims 3 and 7 and

the reference Bass's teaching of message broker (an event message gate unit) (Fig.1, element 108) providing secure message endpoint.

**Referring to claim 33,**

Claim 33 is a claim to the device configured for carrying out the steps of method of claim 4. Therefore, claim 33 is rejected for the reasons set forth for claim 4 and the reference Bass's teaching of message broker (an event message gate unit) (Fig.1, element 108) providing secure message endpoint.

**Referring to claims 37, 45, 46 and 47,**

Claims 37, 45, 46 and 47 are claims to a carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claims 2, 10, 11 and 12. Therefore, Claims 37, 45, 46 and 47 are rejected for the reasons set forth for claims 2, 10, 11 and 12.

**Referring to claims 38 and 42,**

Claims 38 and 42 are claims to a carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claims 3 and 7. Therefore, Claims 38 and 42 are rejected for the reasons set forth for claims 3 and 7.

**Referring to claim 39,**

Claim 39 is a claim to a carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claim 4. Therefore, Claim 39 is rejected for the reasons set forth for claim 4.

**Referring to claims 40 and 41,**

Claims 40 and 41 are claims to a carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claims 5 and 6. Therefore, Claims 40 and 41 are rejected for the reasons set forth for claims 5 and 6.

**Referring to claims 43 and 44,**

Claims 43 and 44 are claims to a carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claims 8 and 9. Therefore, Claims 43 and 44 are rejected for the reasons set forth for claims 8 and 9.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ashok B. Patel whose telephone number is (703) 305-2655. The examiner can normally be reached on 8:00am-5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John A Follansbee can be reached on (703) 305-8498. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Abp  
\*\*\*



ZARNI MAUNG  
PRIMARY EXAMINER